JobManagerv2.0-MSRS-2002-r1.00

IPG Job Manager V2.0

Master Software Requirements Specification

Revision 1.0

01/29/2003
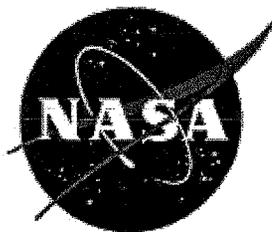
National Aeronautics and Space Administration

Document Information

Document title:	Master Software Requirements Specification
Document file name:	JobManagerMaster2.0_RequirementsSpec_r1.0.doc
Revision number:	2.0
Issued by:	NASA/ARC
Issue Date:	01/29/03
Status:	Final

Document Approvals

	Name	Organization	Title	Signature
Prepared by:	Chaumin Hu	AMTI	Developer	
Reviewed By:	Mark Wallace	AMTI	Developer	
	Warren Smith	CSC	Research Scientist	
	Anthony Lisotta	CSC	Project Manager	
	Kevin Brett	AMTI/HQ	Corporate QA	
	Leigh Ann Tanner	AMTI	Program Manager	
	Piyush Mehrotra	NASA		
	Karen McCann	CSC	Senior Scientist	
	Maurice Yarrow	CSC	Senior Scientist	
	Stu Rogers	NASA	Aerospace Engineer	
	Ken Gee	ELORET		
	John Parks	NASA	Deputy Division Chief	
Approved by:	Thomas Hinke	NASA	Task Requestor	
	William Johnston	NASA	Project Manager	
	William Thigpen	NASA	INE Branch Chief	

IPG Job Manager V2.0
Master Software Requirements Specification

Revision 1.00

01/29/2003

Prepared for

NASA/ARC

By

ADVANCED MANAGEMENT TECHNOLOGY INCORPORATED

And

COMPUTER SCIENCES CORPORATION

Under Contract

DTTS59-99-D-00437/A61812D

Table of Contents

Property of National Aeronautics and Space Administration　　　Working copy if printed

Master Software Requirements Specification

1. Introduction

1.1. Purpose

This specification is intended to provide the environmental and software functional requirements for the IPG Job Manager V2.0 being developed by AMTI for NASA.

1.2. Background and Overview

A basic function of a computational grid such as the NASA Information Power Grid (IPG) is to allow users to execute applications on remote computer systems. The Globus Resource Allocation Manager (GRAM) provides this functionality in the IPG and many other grids at this time. While the functionality provided by GRAM clients is adequate, GRAM does not support useful features such as staging several sets of files, running more than one executable in a single job submission, and maintaining historical information about execution operations.

1.3. General System Description

The purpose of the IPG Job Manager is to allow IPG users to move and copy files between computer systems and to execute applications on specified systems by using Application Programming Interfaces (API) and command line programs. The system shall be accessed only by authorized users and the specified systems shall be within a resource list which is handled by the Job Manager. The Job Manager may be invoked by the IPG Resource Broker, directly by users, or future components such as a workflow manager. The IPG Job Manager shall provide the following functionality:

1. File operations to copy or move files between remote file servers before and/or after one or more applications execute.
2. Execution operations to execute user applications on remote systems.
3. Main sequences to reliably run file and execution operations.
4. Optional cleanup sequences to delete files and directories after a success or a failure of a main sequence.
5. A history of file and execution operations for users to query during or after job execution.
6. Proxy handler to manage the system with the Grid Security Infrastructure for authorized IPG users to access the service.
7. Resource handler to manage a list of resources for users to submit their jobs to the systems within the list.

1.4. Job Execution Model

The IPG Job Manager allows users to specify and run a Job Manager job which consists of a main sequence and an optional cleanup sequence. The main sequence of a job consists of a series of operations. The operation can be File operations (e.g., copying and moving files between machines, creating directories, deleting files and directories, etc.) and Execution operations (executing applications on specified computer systems).

The optional cleanup sequence of a job consists of cleanup subsequences. Each cleanup subsequence consists of a series of file operations. Each cleanup subsequence is associated with an operation in the main sequence.

Figure 1 illustrates the job execution model of the IPG Job Manager system.

When the user submits a job to the Job Manager, the system will start to run the operations of the main sequence in the order in which it was specified by the user. When all operations in the main sequence are executed successfully, the system will run the cleanup subsequence in the reverse order of the operations in the main sequence. For each cleanup subsequence, the system will run the file operations in the order specified by the user.

When a failure occurs during the execution of an operation in the main sequence, no further operations in the main sequence will be executed. Instead, the system will start to run the cleanup subsequence associated with the failed operation and the cleanup subsequences associated with those already executed operations in the reverse order of the operations in the main sequence.

From the point of entry into the cleanup sequence, all file operations will be executed. Even if a file operation fails, the corresponding file's cleanup operations will still be executed.

Property of National Aeronautics and Space Administration   Working copy if printed
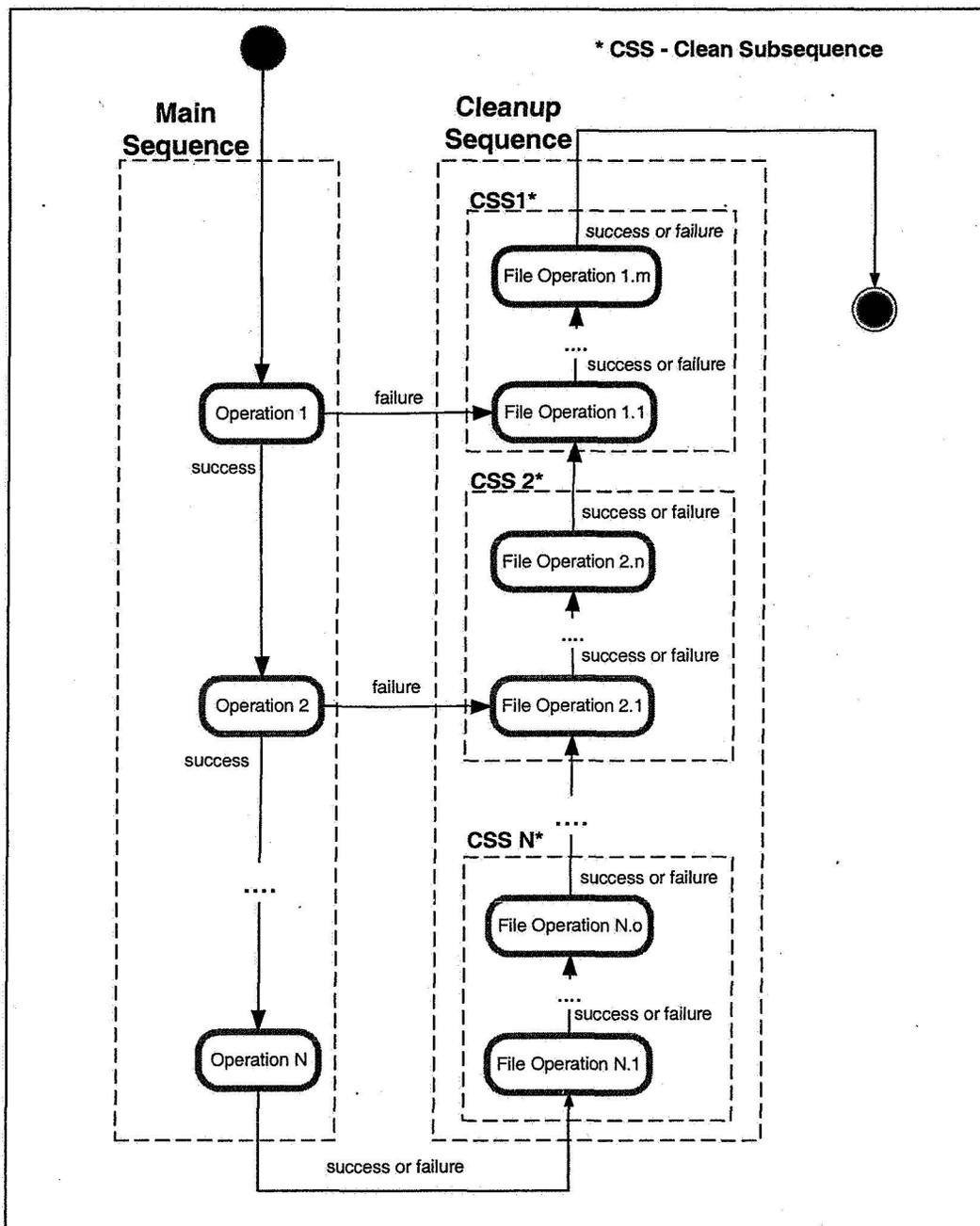
Figure 1. Job Execution Model

1.5. General Requirements

1.5.1. Online Document Documentation

1.5.1.1. Online Programming Documentation

IPG Job Manager application programming interfaces (APIs) and command line programs shall be described in online programming reference documentation for users that wish to use its functionality directly outside of other IPG components.

1.5.1.2. Online User Documentation

IPG Job Manager features and examples of their use shall be described in online user documentation. This documentation shall also provide examples to demonstrate how users can use the Job Manager functions with other packages, like the IPG Resource Broker and the IPG Account Services, etc.

1.5.1.3. Online Administration Documentation

Installation, configuration, and maintenance procedures for the IPG Job Manager shall be described in online administrator documentation.

1.5.2. Error Handling Requirements

1.5.2.1. Error Notification

The Job Manager shall always notify users of errors that are detected.

1.5.2.2. Error Message

The system shall return the name of each error and its error message. The system shall additionally provide the reason for the problem.

1.5.2.3. Error Logging

The Job Manager shall allow users to specify an error log file where error messages will be written to.

1.5.2.4. Timeout Handling

The Job Manager shall provide configurable timeouts to avoid deadlock.

1.5.3. Backward Compatibility

The IPG Job Manager V2.0 cannot maintain 100% backward compatibility with version 1.0. Several APIs and functionality, like the file and execution operations, main and cleanup sequences, etc., are new and different from version 1.0. A document shall be provided for customers who are using version 1.0 of the IPG Job Manager package to easily transform their existing programs to the version 2.0 APIs.

2. Goal Model

2.1. Goals of the System

There are a number of functional goals that the IPG Job Manager System shall satisfy to allow users to submit jobs for staging files and executing applications. These goals reflect the externally visible behavior of the system being developed.

2.2. Goals Hierarchy

The goals hierarchy is an outline of the system's functionality. The specific functions and features of the system, as well as the complete syntax and format of the APIs and command line programs, will be specified in subsequent Build Definition Specifications:

JM1. Register Job Manager Server
JM2. Specify Job
    JM2.1. Specify Main Sequence
        JM2.1.1. Specify File Operation
            JM2.1.1.1. Specify Copy File Operation
            JM2.1.1.2. Specify Move File Operation
            JM2.1.1.3. Specify Delete File Operation
            JM2.1.1.4. Specify Create-Directory File Operation
        JM2.1.2. Specify Execution Operation
    JM2.2. Specify Cleanup Sequence
JM3. Manage Job Execution
    JM3.1. Submit Job
    JM3.2. Assign Job Identifier
    JM3.3. Execute Job
        JM3.3.1. Execute Main Sequence
        JM3.3.2. Execute Cleanup Sequence
    JM3.4. Cancel Job
JM4. Manage Job State
    JM4.1. Assign Job State
    JM4.2. Get Job State
    JM4.3. Notify Job State
JM5. Manage Job Database
    JM5.1. Store Job
    JM5.2. Query Job
    JM5.3. Delete Job
JM6. Administer Job Manager
    JM6.1. Modify User Access

JM6.2. Modify Resource List
JM6.3. Browse Jobs
JM6.4. Log Admin Actions

2.3. Software Requirements

This section describes, at a high level, the software requirements for the system that have been identified in the Goal Hierarchy of the previous section.

JM1. Register Job Manager Server
The Job Manager Server shall be able to register with a registry so that users are able to locate and access it.
Constraints
1. The specific registry utilized for a particular Job Manager shall be specified by the Job Manager Administrator when starting the Job Manager server.
2. The format of the registration and lookup of the Job Manager Server is TBD but shall at the minimum utilize the LDAP protocol and message formats. For the details shall be provided in future documents.
Errors/Exceptions Trapped
1. Invalid authorizations
2. Failure to access registry server
General Error Handling Approach
For invalid authorizations errors, the system shall provide the authorizations that shall be obtained to register the Job Manager server. For failure to access registry server errors, the system shall provide the name of the server that cannot be accessed.

JM2. Specify Job

The user shall be able to specify a job to the Job Manager as given in requirements JM2.1 through JM2.2.

Constraints

1. The system shall allow users to specify a main sequence and an optional cleanup sequence in a Job Manager job.
2. The system shall allow users to specify a job using multiple function calls to specify each operation of the job.
3. The system shall provide an XML format to specify a job in a file and an appropriate function call to construct the job from the file. The XML format shall be designed and described in more detail in the subsequent Build Definition Specifications.

Errors/Exceptions Trapped

The errors/exceptions trapped will consist of those for requirements JM2.1 through JM2.2.

General Error Handling Approach

The general error handling approach will consist of those for requirements JM2.1 through JM2.2.

JM2.1 Specify Main Sequence

The main sequence of a Job Manager job shall consist of file operations (e.g., copying and moving files between machines, creating directories, deleting files and directories, etc.) and execution operations (executing applications on specified computer systems).

Constraints

1. The main sequence shall allow users to specify and run more than one file and execution operation in one Job Manager job.
2. The user shall specify the sequence of the operations in the main sequence.
3. The system shall allow users to specify an optional cleanup subsequence for each operation in the main sequence. The cleanup subsequence shall consist of a series of file operations.

Errors/Exceptions Trapped

1. Invalid syntax
2. Incomplete main sequence

General Error Handling Approach

For all errors, the system shall return the name of the error. For invalid syntax errors, the system shall additionally provide the reason for the syntax violation. For incomplete main sequence errors, the system shall additionally provide the required fields that were missing.

JM2.1.1 Specify File Operation

The system shall allow users to specify various types of file operations in a main sequence.

Constraints

1. The file operation types shall include copy, delete, move, and create directory.
2. The complete attributes of a file operation are listed in section 4.2.1.2.1 (File Operation Data). All attributes are required to specify a file operation.

Errors/Exceptions Trapped

The errors/exceptions trapped will consist of those for requirements JM2.1.1.1 through JM2.1.1.4.

General Error Handling Approach

The general error handling approach will consist of those for requirements JM2.1.1.1 through JM2.1.1.2

JM2.1.1.1 Specify Copy File Operation

The system shall allow users to specify a copy file operation to copy files and directories between computer systems.

Constraints

1. The copy file operation shall provide an option to enable/disable recursive copy for copying a directory tree. The default setting shall be enabled.
2. The permission mode of transferred files shall be carried over with the files from the source system to the destination system.
3. The system shall allow users to specify a filter expression (string pattern with wild card) to copy multiple files that match the expression.

Errors/Exceptions Trapped

1. Invalid syntax
2. Incomplete copy file operation

General Error Handling Approach

For all errors, the system shall return the name of the error. For invalid syntax errors, the system shall additionally provide the syntax violation. For incomplete copy file operation errors, the system shall additionally provide the required fields that were missing.

Property of National Aeronautics and Space Administration

JM2.1.1.2 Specify Move File Operation

The system shall allow users to specify a move file operation to move files between computer systems.

Constraints

1. The move file operation shall provide an option to enable/disable recursive move for moving a directory tree. The default setting shall be enabled.
2. The permission mode of moved files shall be carried over with the files from the source system to the destination system.
3. The system shall allow users to specify a filter expression (string pattern with wild card) to move multiple files that match the expression.

Errors/Exceptions Trapped

1. Invalid syntax
2. Incomplete move file operation

General Error Handling Approach

For all errors, the system shall return the name of the error. For invalid syntax errors, the system shall additionally provide the syntax violation. For incomplete move file operation errors, the system shall additionally provide the required fields that were missing.

JM2.1.1.3 Specify Delete File Operation

The system shall allow users to specify a delete file operation to delete files and directories.

Constraints

1. The delete file operation shall provide an option to enable/disable recursive delete for deleting a directory tree. The default setting shall be enabled.
2. The system shall allow users to specify a filter expression (string pattern with wild card) to delete multiple files that match the expression.

Errors/Exceptions Trapped

1. Invalid syntax
2. Incomplete delete file operation

General Error Handling Approach

For all errors, the system shall return the name of the error. For invalid syntax errors, the system shall additionally provide the syntax violation. For incomplete delete file operation errors, the system shall additionally provide the required fields that were missing.

JM2.1.1.4 Specify Create-Directory Operation

The system shall allow users to specify a create-directory file operation to create directories.

Errors/Exceptions Trapped

1. Invalid syntax
2. Incomplete create-directory file operation

General Error Handling Approach

For all errors, the system shall return the name of the error. For invalid syntax errors, the system shall additionally provide the syntax violation. For incomplete create-directory file operation errors, the system shall additionally provide the required fields that were missing.

JM 2.1.2 Specify Execution Operation

The system shall allow the user to specify executions in a main sequence.

Constraints

1. Required attributes, host name and executable name, shall be specified for an execution operation.
2. The execution operation shall only allow users to specify system-dependent attributes (e.g. absolute directory path, specific host name, etc.) but not system-independent variables (e.g. $SCRATCH1, $HOSTNAME, etc.).
3. The attributes for the execution operation shall allow users to specify a job in enough detail so that the Job Manager can execute it on a specified computer system.
4. The complete attributes of an execution operation are listed in the section 4.2.1.1.2 (Execution Operation Data).

Errors/Exceptions Trapped

1. Invalid syntax
2. Incomplete execution operation

General Error Handling Approach

For all errors, the system shall return the name of the error. For invalid syntax errors, the system shall additionally provide the syntax violation. For incomplete execution operation errors, the system shall additionally provide the required fields that were missing.

JM2.2 Specify Cleanup Sequence

The system shall allow users to specify a cleanup sequence for each job to handle a series of file operations after a success or a failure in the execution of the main sequence.

Constraints

1. The cleanup sequence is optional.

2. The cleanup sequence shall allow users to specify an optional cleanup subsequence associated with an operation in the main sequence. Each cleanup subsequence shall allow users to specify a series of file operations.
3. Each operation in the main sequence shall only have one associated cleanup subsequence.
4. The system shall provide an option to run or not run the cleanup sequence after a failed execution of the main sequence.

Errors/Exceptions Trapped

1. Invalid syntax
2. Incomplete cleanup sequence

General Error Handling Approach

For all errors, the system shall return the name of the error. For invalid syntax errors, the system shall additionally provide the specific violation. For incomplete cleanup sequence

errors, the system shall additionally provide the required fields that were missing.

JM3. Manage Job Execution

The Job Manager shall manage job execution as given in requirements JM3.1 through JM3.4.

Errors/Exceptions Trapped

The errors/exceptions trapped shall consist of those for requirements JM3.1 through JM3.4.

General Error Handling Approach

The general error handling approach shall consist of those for requirements JM3.1 through JM3.4.

JM3.1 Submit Job

The Job Manager shall accept a job submitted by the client.

Constraints

1. A job shall only be accepted if the user has the proper authorizations to access the Job Manager server.
2. The client side of the Job Manager shall verify if the required attributes of the job have been specified before sending the job over to the server side.

Errors/Exceptions Trapped

1. Invalid authorizations to access Job Manager
2. Failure to submit job
3. Incomplete job attributes

General Error Handling Approach

For invalid authorizations to access Job Manager errors, the system shall provide the authorizations that the user shall obtain to submit a job. For failure to submit job error, the system shall provide the name of error and the reason the submission failed. For incomplete attributes errors, the system shall additionally provide the required fields that were missing.

JM3.2 Assign Job Identifier

After a user submits a job to the Job Manager, the Job Manager shall assign and return a job identifier to the user.

Constraints

1. The job identifier shall uniquely identify a job on the IPG.

Errors/Exceptions Trapped

1. Failure to assign job identifier

General Error Handling Approach

For failure to assign job identifier errors, the system shall return the name of the error and additionally provide the reason that it could not assign an ID.

JM3.3 Execute Job
After a user submits a job and a job identifier is assigned, the Job Manager shall execute the main sequence and the optional cleanup sequence (if provided).
Errors/Exceptions Trapped
The errors/exceptions trapped will consist of those for requirements JM3.3.1 through JM3.3.2.
General Error Handling Approach
The general error handling approach will consist of those for requirements JM3.3.1 through JM3.3.2.

JM3.3.1 Execute Main Sequence

The execution of a Job Manager job shall start the main sequence of the job to run the file and execution operations that were specified by the user.

Constraints

1. The system shall run the main sequence first before the cleanup sequence.
2. A main sequence shall only be executed if the user has the proper authorizations to do so on the target systems.
3. All operations shall be run in the order that was specified by the user.
4. The system shall run the file operations using grid file transfer mechanisms. Two examples of these mechanisms are the grid enabled SCP and FTP utilities that have been enhanced with Grid Security Infrastructure (GSI) security mechanisms.
5. The execution operations shall be performed using the GRAM utility.

Errors/Exceptions Trapped

1. Failure to run file or execution operation
2. Invalid authorizations to access target systems
3. Error detected from GRAM service (failure to contact host, invalid attributes, failure to create stderr file, cannot find executable, etc.)
4. Error detected from GridFTP service (cannot find file, invalid directory, invalid permission to write file, etc.)
5. Recoverable file operation error detected from GridFTP service (connection failure to stage file, processing reply error, etc.)

General Error Handling Approach

For any failure that is detected during the running of the main sequence, the system shall stop the main sequence and jump to the cleanup sequence. The cleanup sequence shall start to run the cleanup subsequence associated with the failed operation first and then the cleanup subsequences associated with those already executed operations in the reverse order of the operations in the main sequence.

For invalid authorizations to access target systems errors, the system shall provide the authorizations that the user must obtain to run a job. For all errors from GRAM and GridFTP services, the system shall provide the error names and the error messages provided by the services to the users. For recoverable file operation errors, the system shall provide a retry mechanism to attempt to re-run the file operation. This mechanism shall be designed and described in more detail in the subsequent Build Definition Specifications.

JM3.3.2 Execute Cleanup Sequence

After successfully running the main sequence or when a failure is detected during a run of the main sequence, the system shall run the cleanup sequence (if provided and activated) to execute the file operations in the cleanup sequence that were specified by the user.

Constraints

1. When executing a cleanup sequence after successfully running the main sequence, the system shall run the cleanup subsequence in the reverse order of the operations in the main sequence.
2. When executing a cleanup sequence after a failed operation in the main sequence, the system shall start to run the cleanup subsequence associated with the failed operation and then the cleanup subsequences associated with those already executed operations in the reverse order of the operations in the main sequence.
3. To execute file operations in each cleanup subsequence, the system shall run the file operations in the order specified by the user.
4. From the point of entry into the cleanup sequence, all file operations shall be executed. Even if a file operation fails, the corresponding file's cleanup operation will still be executed.
5. The cleanup sequence shall only be executed if the user has the proper authorizations to do so on the target systems.

Errors/Exceptions Trapped

1. Failure to execute file operation in the cleanup subsequence.
2. Invalid authorizations to access target systems

General Error Handling Approach

For failure to execute file operation errors, the system shall notify the user of the error and then keep executing the subsequent file operations. For invalid authorizations to access target systems errors, the system shall provide the authorizations that the user must obtain to run the clean up sequence.

JM3.4 Cancel Job

Users shall be able to cancel the execution of a job that they have previously submitted to the IPG Job Manager.

Constraints

1. The system shall provide an option to run or not run the cleanup sequence after the cancel operation.
2. To execute the cleanup sequence after a cancel operation, the system shall run the cleanup subsequence associated with the main operation first at the point of the cancel, and then the cleanup subsequences associated with those already executed operations in the reverse order of the operations in the main sequence.
3. The canceled job shall be deleted from the Job Manager database.

Property of National Aeronautics and Space Administration   Working copy if printed

Errors/Exceptions Trapped
1. Failure to cancel job
General Error Handling Approach
For failure to cancel job errors, the system shall provide the reason that the job could not be canceled.

JM4. Manage Job State

The Job manager shall manage job states as given in requirements JM4.1 through JM4.3.

Errors/Exceptions Trapped

The errors/exceptions trapped shall consist of those for requirements JM4.1 through JM4.3.

General Error Handling Approach

The general error handling approach shall consist of those for requirements JM4.1 through JM4.3.

JM4.1 Assign Job States

The Job Manager shall assign and keep various job state events of a running job and allow users to retrieve these states.

Constraints

1. The job state events of a job shall include the start and complete times of each operation, the execution state events for execution operations (submitted, waiting, running, done, etc) provided by the GRAM utility, the file state events for file operations (copy, move, delete, create directory, etc), and any error event that occurs during the job execution.
2. Beside simple state events, the system shall be extensible to provide new events and additional detailed information for each job state to the users.
3. The system shall assign the current state and the state history to a job.

Errors/Exceptions Trapped

1. Failure to assign job state

General Error Handling Approach

For failure to assign job state errors, the system shall provide the reason that the state could not be assigned.

JM4.2 Get Job State

A user shall be able to retrieve the state of a job during or after job execution.

Constraints

1. The system shall allow users to get the current state as well as the history of job states.

Errors/Exceptions Trapped

1. Failure to get job state

General Error Handling Approach

For failure to get job state errors, the system shall provide the reason why the job state was not obtained.

JM4.3 Notify Job State

A user shall be able to request that the Job Manager notify them as a job changes state.

Constraints

1. The system shall provide a mechanism for users to register for notification. The mechanism shall be designed and described in more detail in the subsequent Build Definition Specifications.
2. The system shall notify any clients that registered for notification of the state changes of the job.
3. The system shall provide the option for users to stop sending the state change notification.

Errors/Exceptions Trapped

1. Failure to notify user of job state changes

General Error Handling Approach

For failure to notify user of job state changes errors, the system shall provide the reason why the user could not be notified.

JM5. Manage Job Database

A Job Database shall be available that satisfies requirements JM5.1 through JM5.3.

Errors/Exceptions Trapped

The errors/exceptions trapped will consist of those for requirements JM5.1 through JM5.3.

General Error Handling Approach

The general error handling approach will consist of those for requirements JM5.1 through JM5.3.

JM5.1 Store Job

The Job Database shall be able to store information about jobs that have been submitted to the Job Manager.

Constraints

1. Job information shall include the information originally specified by the user, the current state and the state history of the job.
2. A user shall be able to configure the period of time that job information will be stored.

Errors/Exceptions Trapped

1. Failure to store job

General Error Handling Approach

For all errors, the system shall return the name of the error. For failure to store job errors, the system shall additionally provide the reason why the job could not be stored.

JM5.2 Query Job

A user shall be able to retrieve information about a job from the Job Manager Database based on combinations of job characteristics such as user, job identifier, target system, current state, etc.

Constraints

1. Job information shall include the information originally specified by the user, the current state and the state history of the job.
2. Users shall not be able to retrieve information about the jobs of other users.

Errors/Exceptions Trapped

1. Failure to query job
2. Invalid job characteristics

General Error Handling Approach

For all errors, the system shall return the name of the error. For failure to query job errors, the system shall additionally provide the reason why the job could not be queried. For invalid job characteristics errors, the system shall additionally provide the invalid job characteristics.

Property of National Aeronautics and Space Administration   Working copy if printed

JM5.3 Delete Job

The Job Manager shall allow users to delete a job from the database based on job identifier.

Constraints

1. Users shall not be able to delete jobs of other users.

Errors/Exceptions Trapped

1. Invalid job identifier
2. Invalid authorizations

General Error Handling Approach

For all errors, the system shall return the name of error. For invalid authorizations errors, the system shall additionally provide the authorizations that the user shall obtain to remove the specified job.

JM6. Administer Job Manager

A set of administration command-line programs shall be available that satisfies requirements JM6.1 through JM6.4.

Errors/Exceptions Trapped

The error/exceptions trapped will consist of those for requirements JM6.1 through JM6.4.

General Error Handling Approach

The general error handling approach will consist of those for requirements JM6.1 through JM6.4.

JM6.1 Specify User Access

A command-line program shall be provided for a Job Manager administrator to specify who can use the Job Manager.

Constraints

1. The program shall only be used if the user has the authorizations to do so.
2. The actual modification will be performed by the IPG Job Manager server by setting up the user list.
3. The program shall provide an option for local and remote administrators to display the user list.

Errors/Exceptions Trapped

1. Invalid authorizations
2. Failure to connect to Job Manager
3. Failure to specify user access

General Error Handling Approach

For all errors, the system shall return the name of the error. For invalid authorizations errors, the system shall provide the reason why authorization was not granted. For failure to connect to Job Manager errors, the system shall provide the name of the server that could not be contacted and the reason the connection failed. For failure to specify user access errors, the system shall provide the reason that the user access could not be set.

JM6.2 Specify Resource List

A command-line program shall be provided for a Job Manager administrator to specify where the Job Manager can submit jobs.

Constraints

1. The program shall only be used if the user has the authorizations to do so.
2. The actual modification will be performed by the IPG Job Manager server by setting up the resource list.
3. The program shall provide an option for local and remote administrators to display the resource list.

Errors/Exceptions Trapped

1. Invalid authorizations
2. Failure to connect to Job Manager
3. Failure to specify resource list

General Error Handling Approach

For all errors, the system shall return the name of the error. For invalid authorizations errors, the system shall provide the reason why authorization was not granted. For failure to connect to Job Manager errors, the system shall provide the name of the server that could not be contacted and the reason the connection failed. For failure to specify resource list errors, the system shall provide the reason that the resource list could not be modified.

JM6.3 Browse Jobs

A command-line program shall be provided for a Job Manager administrator to browse all of the jobs in the job database.

Constraints

1. The program shall only be used if the user has the authorizations to do so.
2. The program shall provide an option for local and remote administrators to browse jobs.

Errors/Exceptions Trapped

1. Invalid authorizations
2. Failure to connect to Job Manager
3. Failure to browse jobs

General Error Handling Approach

For all errors, the system shall return the name of the error. For invalid authorizations errors, the system shall provide the reason why authorization was not granted. For failure to connect to Job Manager errors, the system shall provide the name of the server that could not be contacted and the reason the connection failed. For failure to browse jobs errors, the system shall provide the reason that jobs could not be browsed.

JM6.4 Log Admin Actions

A logging capability shall be provided to log all administrative actions.

Constraints

1. The system shall provide an option to enable or disable the logging feature.
2. The system shall allow the user to specify the log name.

Errors/Exceptions Trapped

1. Invalid authorizations
2. Failure to log actions

General Error Handling Approach

For all errors, the system shall return the name of the error. For invalid authorizations errors, the system shall provide the reason why authorization was not granted. For failure to log actions errors, the system shall provide the reason that the actions cannot be logged.

Property of National Aeronautics and Space Administration

3. Environmental Model

This section describes the IPG Job Manager System non-behavioral requirements, i.e., requirements that affect the entire system or have system wide applicability, but do not describe specific system functions.

3.1. Elements

There are two main elements that shall be constructed to satisfy the functionality described in Section 2. These elements are a job manager, which is responsible for accepting, executing, and monitoring jobs, and a job manager database, which stores jobs that have been specified and submitted by users.

3.2. Security and Permissions Model

The table below describes the various types of users, the modules or subsystems that they are allowed to access and the functions that they are allowed to perform within those modules.

3.2.1. Job Manager

User Type	Job Manager
Admin	Start and stop Job Manager
Client Application	Submit*, query*, and cancel* jobs
<Note: * Users can only access their own Job Manager jobs>	

3.2.2. Job Manager Database

User Type	Job Manager Database
Admin	Start and stop database Read, write, and delete jobs
Job Manager	Read, write and delete jobs
Client Application	Read*, write*, and delete* jobs
<Note: * Users can only access their own Job Manager jobs>	

3.3. Traceability

The software functional requirements presented in this specification are traceable back to the "IPG Job Manager Requirement List" document that was finalized on 10/15/2002. Detailed tracing of these requirements is provided in Appendix B.

3.4. Standards Compliance

3.4.1. Coding Standards Compliance

The IPG Job Manager shall be designed in accordance with the following references to develop programs and provide APIs:

1. "Code Conventions for the Java Programming Language" by Sun Microsystems http://java.sun.com/docs/codeconv.
2. "The Elements of Java Style" by Allan Vermeulen.
3. "C++ Coding Standard" http://www.possibility.com/Cpp/CppCodingStandard.html. .

3.5. General User Interface or End-User Requirements

The IPG Job Manager shall have an API interface for C/C++, Java, Perl, and Python, as well as a command line interface. These APIs and the command line interface shall be available under the IRIX, Linux, and Solaris operating systems. No hardware requirements have been specified. The command-line Job Manager interface shall provide optional arguments for users to specify, submit, cancel, and monitor jobs. An XML-based argument file format shall also be provided by the system for users to specify jobs and also to be read by the command line Job Manager programs.

3.6. Application Development Environment

The requirements specified in Section 3.5 indicate that the IPG Job manager shall be developed and tested under the IRIX, Linux, and Solaris operating systems.

3.7. Design Constraints

3.7.1. Client-Server Framework Constraint

The IPG Job Manager shall be designed under a framework provided by the CODE (Control and Observation in Distributed Environments) package to handle its events and communication between the client and server sides of the Job Manager.

Property of National Aeronautics and Space Administration   Working copy if printed

3.7.2. Software Constraints

The IPG Job Manager shall be designed and developed using the following software packages:

1. Globus Java CoG Kit client library.
2. Globus Java GridFTP client library (jftp).
3. CODE framework package.

3.8. Server/Mainframe Software/Hardware Requirements

3.8.1. Software Requirements

The IPG Job Manager shall be designed and developed using the following server/mainframe software:

1. Globus.
2. GRAM job manager server.
3. GridFTP server.

3.8.2. Hardware Requirements

None specified.

3.9. Database and Data Management

3.9.1. Archiving

Data shall be archived and backed up according to the file system archiving policy where the IPG Job Manager is deployed. The period for the database to hold the job data shall be decided later after testing under load.

3.9.2. Data Retention

Archival data shall be retained according to the file system data retention policy where the IPG Job Manager is deployed.

3.10. Accessibility Factors

The software shall be designed and developed to be compliant with the Government Section 508 requirements for "Software Applications and Operating Systems" as documented in CFR 1194.21."

3.11. System Software Environment

The IPG Job Manager shall be developed using a client-server architecture. The server shall use the Globus GRAM service to execute applications on remote computer systems and the GridFTP service to stage files on remote computer systems. A Grid Information Service (GIS) registry shall be used so that IPG Job Manager client can locate servers. This environment is shown in Figure .
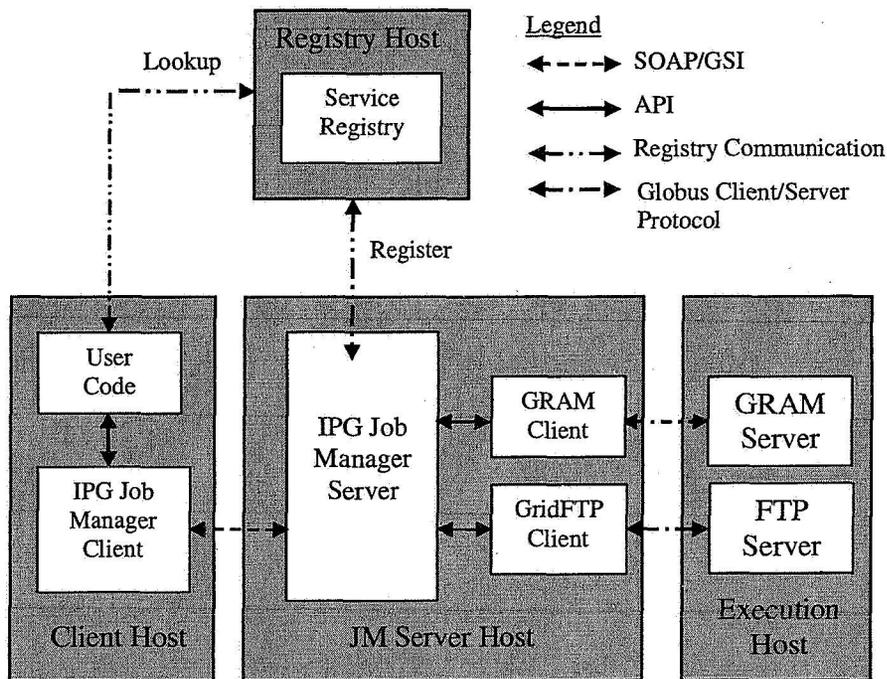
Figure 2. System Software Environment

Property of National Aeronautics and Space Administration

4. Data Model

4.1. External Interfaces

The IPG Job Manager shall be used by IPG client applications. Changes to the IPG Job Manager interface will affect these applications. Figure shows that the IPG Job Manager uses the GRAM and GridFTP client interfaces and interacts with a service registry.

System Input			
Source: <System or Interface Name>	**Data Entity Name**	**Data Type**	**Type of Connection or Communication**
IPG client applications	inputJob	IPG JM job	API

System Output			
Destination: <System or Interface Name>	**Data Entity Name**	**Data Type**	**Type of Connection or Communication**
IPG client applications	jmState	IPG JM state	API
Service Registry	registryCode	Register code	Registry protocol
Globus GRAM	gramJob	GramJob	API
Globus GridFTP	ftpClient	FtpClient	API

4.2. Data Description

The IPG Job Manager maintains a database of jobs. The major entries in this database include IPG Job Manager jobs and service registrations. Each Job Manager job contains a main sequence and an optional cleanup sequence. The main sequence contains file and execution operations. The cleanup sequence contains delete file operations.

4.2.1. Job Manager Job Data

Comprised of:	A main sequence and an optional cleanup sequence as given in 4.2.1.1 and 4.2.1.2.
Derived from:	Manual entry by users.
Used for:	Describing a Job Manager job.
Accessible by:	**Users** – read/write of jobs that they have created. **Administrators** – read/write of all jobs.

Property of National Aeronautics and Space Administration   Working copy if printed

4.2.1.1. User Proxy Data

Comprised of:	The proxies of users to be forwarded from the client side to the server side of the IPG Job Manager.
Derived from:	IPG Job Manager client-side package.
Used for:	Authenticating users to IPG resources.
Accessible by:	**IPG Job Manager** – read/write proxy data.

4.2.1.2. Main Sequence Data

Comprised of:	The sequence of file and execution operations to be executed as given in 4.2.1.2.1 and 4.2.1.2.2.
Derived from:	Manual entry by users.
Used for:	Describing the main sequence of a Job Manager job.
Accessible by:	**Users** – read/write of sequences that they have created. **Administrators** – read/write of all sequences.

4.2.1.2.1. File Operation Data

Comprised of:	1. File operation type (copy, move, delete, and create-directory) 2. Source host name 3. Source directory 4. Source file name or filtering expression 5. Destination host name * 6. Destination directory * 7. Destination file name or filtering expression * 8. Recursive flag (not for create-directory operation type) 9. Permission carry-over flag * [Note: * For copy and move operation type only]
Derived from:	Manual entry by users.
Used for:	Describing a file operation with specified file operation type, source and destination remote systems, file name, and recursive flag, etc.
Accessible by:	**Users** – read/write of attributes of the file operation. **Administrators** – read/write of all attributes of the file operation.

4.2.1.2.2. Execution Operation Data

Comprised of:	1. Host name 2. Executable directory 3. Executable name 4. Executable arguments 5. Maximum amount of memory required 6. CPU time in minute units 7. Wall time in minute units 8. Queue name 9. Project name 10. Delegation type 11. Standard input 12. Standard output 13. Standard error 14. Environment variables 15. Job start up type 16. Number of CPUs
Derived from:	Manual entry by users.
Used for:	Describing the application to be executed on a computer system.
Accessible by:	**Users** – read/write of attributes for the execution operation that they have created. **Administrators** – read/write of all execution operations.

4.2.1.3. Cleanup Sequence Data

Comprised of:	Cleanup subsequences which are comprised of a series of file operations.
Derived from:	Manual entry by users.
Used for:	Describing a cleanup sequence.
Accessible by:	**Users** – read/write of cleanup sequence that they have created. **Administrators** – read/write of all cleanup sequences.

4.2.1.4. Job State History Data

Comprised of:	Information about the state changes history of a job.
Derived from:	IPG Job Manager.
Used for:	Describing the state changes history of a job.
Accessible by:	**Users** – read only. **Administrators** – read/write.

4.2.1.4.1. Job State Data

Comprised of:	Information about one state in the state history of a job. 1. State type (submitted, waiting, running, done, etc) 2. State change time
Derived from:	IPG Job Manager.
Used for:	Describing one state in the state history of a job.
Accessible by:	**Users** – read only. **Administrators** – read/write.

4.2.1.5. Job Identifier Data

Comprised of:	Information about the job identifier of a job.
Derived from:	Issued by Job Manager after a specified job had been submitted to the Job Manager.
Used for:	Users and administrators to query job in the database.
Accessible by:	**Users** – read. **Administrators** – read. **IPG Job Manager** – write.

4.2.2. Service Registration Data

Comprised of:	Information about the location of an IPG Job Manager server, the protocol to use to talk to an IPG Job Manager server, and the interface of the IPG Job Manager service.
Derived from:	The location and description provided by an IPG Job Manager server.
Used for:	Locating IPG Job Manager servers.
Accessible by:	**Users** – read only. **IPG Job Manager Servers** – read/write.

4.3. Relationships

There is no relationship between the data entities of the Job Manager job and the service registration data.

4.3.1. Data Constraints

None specified.

4.3.2. Message Formats

In the previous Figure , the communication between the client and server of all subsystems is to take place using a GSI compliant interface that uses data formatted in the eXtensible Markup Language (XML) using Simple Object Protocol (SOAP) as the transport mechanism. The registration and lookup of the Job Manager Server will utilize the LDAP protocol and message formats.

4.3.3. Logical DBMS Model

The exact database model and implementation for the database described in sections 4.2.1 and 4.2.2 have yet to be determined and will be specified in subsequent Build Definition Specifications.

4.4. Data Dictionary

The exact tables and data items included in the database described in sections 4.2.1 and 4.2.2 have yet to be determined and will be specified in subsequent Build Definition Specifications.

4.5. Data Conversion Rules

There is no existing data to convert so there are no applicable rules.

5. Operations Model

This section describes the processes or personnel responsible for maintenance and operation of the system, database, security, network or other aspects of the system such as backup, recovery, and archiving.

5.1. Operations Modes and Major Functional Entities

There are two modes of operation for this system.

5.1.1. User Mode

In this mode of operation, users can contact a registry to locate an IPG Job Manager Server and can use the IPG Job Manager client API to connect to it. Users can then use the client APIs to specify, submit, monitor, cancel, and lookup jobs.

Users can also use the Job Manager command-line programs to perform all functions provided by the API.

5.1.2. Administration Mode

This mode of operation will be used by system administrators to maintain the software. The main functionality that is exposed is described in section 2.3 under requirements JM6.1 (Specify User Access), JM6.2 (Specify Resource List), JM6.3 (Browse Jobs), and JM6.4 (Log Admin Actions).

5.2. Server and System Operations

This section describes the processes for maintenance and operation of the system, database, security, network or other aspects of the system such as backup, recovery, and archiving.

5.2.1. IPG Job Manager Administrator

One or more administrators are responsible for installing, maintaining, and ensuring the correct operation of each IPG Job Manager Server.

5.2.2. System Administrator

System administrators shall be responsible for maintaining the server hardware, operating system, and other software. These administrators may or may not be the same as the IPG Job Manager administrators.

5.2.3. Software Repository Administrator

One or more administrators are responsible for installing, maintaining, and ensuring the correct operation of each Software Repository server.

5.3. Bandwidth and Database and Transaction Capacity Requirements

5.3.1. Performance Requirements

5.3.1.1. Reply to Job Submit.

A Job Manager server shall reply to requests in number of seconds which is TBD during the implementation.

5.3.1.2. Scalability

The Job Manager shall be designed so that it is able to scale up to a large number of system resources and end users. The number of resources and end users are TBD and shall be evaluated and defined during the implementation.

Appendix A. Glossary of Terms

API – Application Programming Interface

CODE – Control and Observation in Distributed Environments

GRAM – Globus Resource Allocation Manager

GridFTP – Grid File Transfer Protocol

GSI – Grid Security Infrastructure

IPG – Information Power Grid

JM – Job Manager

LDAP – Lightweight Directory Access Protocol

RPC – Remote Procedure Call

SOAP – Simple Object Access Protocol

XML – eXtensible Markup Language

Appendix B. Requirements Traceability

The appendix traces the requirements specified in this document back to the requirements specified in the *IPG Job Manager Requirements List* document that was finalized on 10/15/02.

<Note: The [CX] notations, where X is an integer, in the table bellow, represent the future options, which are defined in Appendix C.>

IPG Requirement List, 10/15/2002	MSRS	How satisfied
3.1 Register JM server	JM1	Fully
3.2 Job Model		Partially
3.2.1 Single System Job	JM2	Fully
3.2.2. Sequence Job	JM2	Fully
3.2.3 Co-Scheduling Job	[C1]	No
3.2.4 Workflow Job	[C2]	No
3.3 Specify Job		Partially
3.3.1. Specify Job Attribute	JM2	Fully
3.3.2. Further/User-Defined Attributes	[C15]	No
3.4 Execute Job	JM3.3	Fully
3.5 Assign Unique Identifier	JM3.2	Fully
3.6 Stage File / File Operation	JM 2.1.1	Fully
3.6.3 Stage File In Parallel	[C3]	No
3.6.4 Recursive File Transfer	JM2.1.1	Fully
3.6.5 Wildcards in File Name	JM2.1.1	Fully
3.6.6. Stage File in Different System	[C4]	No
3.6.7.1 File Permission Mode	JM2.1.1	Fully
3.6.7.2 Enable/Disable Mode Transfer	JM2.1.1	Fully
3.6.7.3 Access Permission Mode	[C5.1]	No
3.6.7.4 Chmod Remote Files	[C5.2]	No
3.6.7.5 Mode for different system	[C5.3]	No
3.6.8 Create Directory	JM2.1.1.4	Fully
3.6.9 Delete Files	JM2.1.1.3	Fully
3.6.9.1 Delete File & Directory	JM2.1.1.3	Fully

Property of National Aeronautics and Space Administration   Working copy if printed

3.6.9.2 Recursive delete	JM2.1.1.3	Fully
3.6.9.3 Delete generated file	JM2.1.1.3	Fully
3.6.9.4 Delete filtering files	JM2.1.1.3	Fully
3.6.10 Rename File	JM2.1.1.2	Fully
3.6.11 Move Files	JM 2.1.1.2	Fully
3.6.12 File Caching	[C6]	No
3.6.13 Multiple File Operation in one Job	JM2.1	Fully
3.7 Stop/Restart Job	JM3.4	Fully
3.8 Pause Job	[C8]	No
3.9 Cancel Job (with clean up features)	JM3.4	Fully
3.10 Get Job	JM5.2	Fully
3.11 Software Dependency	IPG Portability Manager	Fully
3.13 Directory Structure 3.13.1 Variable Substitution 3.13.2 Scratch Space	IPG Portability Manager	Fully
3.13 Job State 3.13.1 Basic Job State 3.13.2 Further State 3.13.3 Get Job State 3.13.4 Job State Notification	JM4 JM4.1 JM4.1 JM4.2 JM4.3	Fully Fully Fully Fully Fully
3.14 Error Handling 3.14.1 Pre-Check Job Setup 3.14.2 Pre-Check System 3.14.3 Error Notification 3.14.4 Error Message 3.14.5 Error Logging 3.14.6 Restart Job Automatically 3.14.7 Timeout Handling	1.4.2 JM3.3.1 [C17] 1.4.2.1 1.4.2.2 1.4.2.3 [C10] 1.4.2.4	Partially Fully No Fully Fully Fully No Fully
3.15. Job Database 3.15.1 Store Job 3.15.2 Query Job 3.15.3 Searchable	JM5 JM5.1 JM5.2 JM5.2	Partially Fully Fully Fully

Working copy if printed

3.15.4 Data Privacy	JM5.2	Fully
3.15.5 Data Sharing	[C17]	No
3.15.6 Data Integrity	[C18]	No
3.16 Cost Payment Consideration	[C11]	No
3.17 Setup Environment	IPG Portability Manager	Fully
3.18 Administration	JM6	Fully
3.18.1 System Administration	JM6.1 & JM6.2	Fully
3.18.2 Remote Administration	JM6.2 & JM6.2	Fully
3.18.2 Administration Logging	JM6.3	Fully
3.9 Interface		Partial
3.9.1 Java API	3.5	Fully
3.9.2 C/C++ API	3.5	Fully
3.9.3 Perl API	3.5	Fully
3.9.4 Python API	3.5	Fully
3.9.5 Command-Line Program	3.5	Fully
3.9.6 Java GUI	[C12]	No
3.9.7 Java Bean/Servlet	[C13]	No
3.20 Performance and Scalability	5.3	Fully
3.20.1 Reply to Job Submit	5.3.1.1	Fully
3.20.2 Scalability	5.3.1.2	Fully
3.21 Availability (for non-NASA users)	[C14]	No
3.22 Documentation	1.4.1	Fully

Appendix C. Future Requirements

This appendix lists the requirements which were included in the IPG Job Manager Requirements List, 10/15/2002 but not be included in the V2.0 release.

These requirements will be considered in the future after Job Manager V2.0.

[C1]   Co-Scheduling Job

The Job Manager shall execute a single application on, say, Chapman and Lomax. The applications on each machine would run simultaneously and might communicate using MPICH-G2 or CORBA. This might be useful for multi-disciplinary applications.

[C2]   Workflow Job

1. The Job Manager could be given a Directed Acyclic Graph of file and compute operations and it shall be able to execute these operations.
2. Job Manager is essentially a work flow manager. A job can be a workflow batch script, which is embedded with the specific system architecture and policy related definition.
3. It is required to provide a language essentially allowing users to define the work flow of the job execution. The Job Manager user can use the language to specify the work flow of the job execution in parallel, in series and/or with the tasks defined in the way of the Directed Acyclic Graph.
4. An Execution Process Transfer module may be required to convert a system-neutral job workflow description (Abstract Job Object) to a system-dependent job batch file. The system related issues include the specific architecture and policy. This approach provides a consistent API for user but allows the server to work dynamically with different types of systems.
5. Job Manager shall provide a GUI tool together with the workflow model, allowing users to draw the work flow for job execution.

Note: the final decision to provide the feature of Workflow Job Model is TBD at this moment, and shall be further investigated and provided in another document

[C3]   Stage Files in Parallel

For staging multiple files, the Job Manager shall allow the user to choose whether to stage files in sequence or parallel.

[C4]   Stage Files in Different Type of System

The Job Manager shall allow user to stage files between different type of computer and operation systems. For example, Window 2000 and UNIX.

[C5] Access File Permissions Mode

1. The user shall be able to access the permissions mode of the remote file.
2. The user shall be able to change (chmod) the permission mode of the remote files.
3. For staging files between different types of the systems, the Job Manager shall provide a proper mechanism to handle the permission mode transformation.

[C6] File Caching

1. File caching mechanism shall be provided to improve the performance of the job execution.
2. The Job Manager shall allow users to specify the cache directories and files in the remote machine.
3. The Job Manager shall automatically check the existence of the required files in the cache to determine if the files need to be copied.
4. The Job Manager shall allow user to decide whether the cached files shall be deleted or not after job execution.

Note: Final decision to provide the feature of File Caching is TBD at this moment since it requires further investigation.

[C7] Pause Job

Users shall be able to ask the Job Manager to pause the job and then continue the execution.

[C8] Directory Structure

The user shall not have to know the directory structure of the system that they want to use. There are several ways this can be addressed, and they could be addressed here or in the broker.

[C9] Variable Substitution

1. A variable such as $home can be specified by the users in their directory paths. This string will be substituted with the users' actual home directories on the machine.
2. A variable such as $scratch can be specified by the users in their directory paths.
3. A variable such as $hostname or $exechost can be substituted with the name of the host the application will execute on.

[C10] Restart Job Automatically

For any recoverable error detected, the Job Manager shall be able to restart the job automatically. The user shall be able to enable and to disable this feature.

[C11] Cost Payment Consideration

The design of the Job Manager shall consider who will pay the cost of the job execution.

The Job Manager shall find a strategy and mechanism to minimize the cost of job execution. Specific strategies and mechanism are TBD at this moment, and shall be further investigated and provided in another document.

[C12] Java GUI
GUI tools shall be provided for demonstrations and convenience of those users who like GUIs. Further details shall be provided in a separate document.

[C13] Java Bean/Servlet for Plugging into Portals
A Java Bean/Servlet package shall be provided for users to create their own application or scientific Portals. Detailed specification shall be provided separately.

[C14] Availability
IPG Job Manager package shall be made available to those non-NASA and non-US Government users. Making our software widely available is critical to its success in the grid community.

[C15] User-Defined Job Attributes
Users shall be able to attach arbitrary attributes to jobs, for example, the information about the simulation method, simulation parameters, and simulation results.

[C16] Pre-Check System
To optimize the system usage, the Job Manager shall pre-verify the availability of resource for job execution, like if the system is up and/or if the system is in dedicated time, etc., of the specified systems before submitting and executing the job.

[C17] Data Sharing
Users should be members of groups and job information should be made accessible to the group.

[C18] Data Integrity
When users start adding notes to their runs in the database, the data shall be backed up so that it can be quickly restored.